Assignment Report: Fruit Calories Program

Deadline: 11:59PM on 14 Feb 2024

1. Introduction:

This assignment is to create a Python program that takes a fruit name as input and prints the calories present in one portion according to the FDA's fruit calories poster, as provided in the assignment's instruction letter.

2. Problem Statement:

The U.S. Food & Drug Administration (FDA) provides nutrition information for the 20 most frequently consumed raw fruits in the United States. Your task is to implement a Python program, that allows consumers to input a fruit (case-insensitively) and outputs the number of calories in one portion of that fruit according to the FDA's poster. You can find the poster here https://www.fda.gov/media/76508/download?attachment

3. Methodology:

3.1 Data Representation:

A "*dictionary*" named `calories` was used to store the calories for all 20 fruits. Each fruit's name was the key and its corresponding calorie count for one portion, as per the FDA's poster was the value.

```python
calories = {
    "apple" : 130,
    "avocado" : 50
    # ... (entries for other fruits)
}
```

3.2 User Input Handling:

The program prompts users to enter a fruit, converts the input to lowercase for case-insensitivity, and checks if the entered fruit is a key in the dictionary, using "*in*" operator. The result is then printed to the console as "*formatted string literal*".

```python
user_input = input("Enter a fruit: ").lower()

if user_input in calories:
    calorie_in_fruit = calories[user_input]
    print(f"Accoring to FDA's Poster, The calories in one portion of
{user_input.capitalize()} are {calorie_in_fruit}.")
else:
    print("Invalid input. Please enter a valid fruit.")
```

4. Implementation:

A variable "calories" is made of type _dictionary_, it stores names and calories of food in key-value pairs. Names of food are stored in lower case to support the case-insensitivity feature of the program
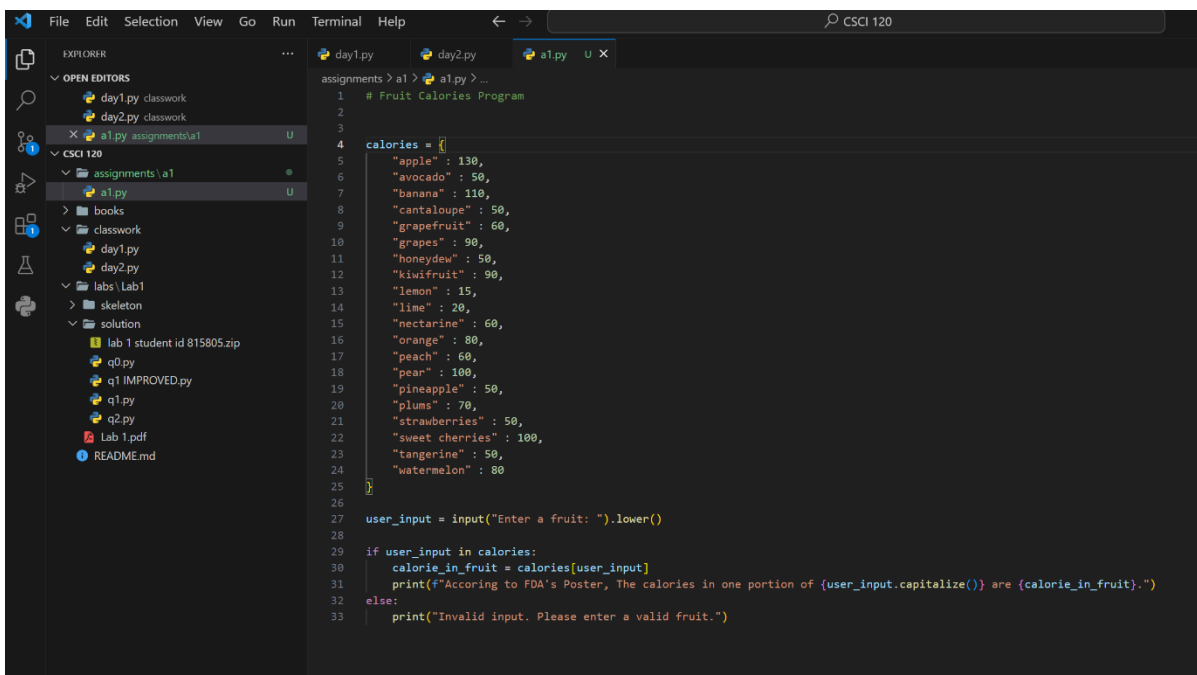
Then, an input is shown to the user using "_input()_" function, with argument of the prompt which will be shown to the user and the value that user entered is stores in a variable "user_input", after using "_.lower()_" of string to make the text in lower cast, which will support the case-insensitivity feature, as the keys of the variable "calories" (Fruit Names) are already in lower case.

After that, control flow statement is implemented using "_if_" checks weather the "user_input" is in "calories" dictionary using membership operator "_in_". If it is in the dictionary, it saves the calories of the fruit that user inputted in a variable "calorie_in_fruit" using _Index brackets "[]"_ then it prints the calories of that fruit's one portion using a "_formatted string literal_". If the "user_input" is not in "calories" dictionary, the program prints an error message.
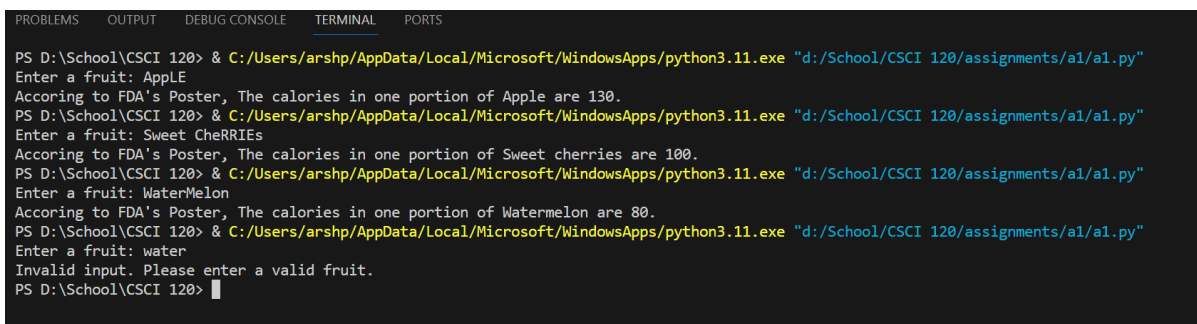
5. Results:

The program does everything as mentioned assignment, providing users with calorie information for the entered fruit with case-insensitive handling of the user's input.

6. Screenshots:

Figure 1: Screenshot of the source code of the program.

Figure 2: Screenshot of the execution of the program.

7. Efficiency Considerations:

Case-Insensitive Handling: By storing all fruit names in lowercase within the dictionary and converting user's input to lowercase, the program supports case-insensitive input. This ensures that users can input fruit names in any casing without affecting the program's functionality.

Dictionary Lookup: The use of a dictionary for storing calorie information allows for efficient approach to get calorie of specific fruit using index brackets, instead of using for/while loops. This approach reduces execution time, contributing to the program's efficiency and it will also make the code cleaner.

Membership Operator: The use of a "*in*" in "*if*" statement to check if the user input exists in the dictionary allows more readability of code, instead of using other methods like using exception handling, using get() method, etc.

8. Acknowledgments:

I would like to acknowledge the collaborative effort of me and my peers in the completion of this assignment. Arshpreet Singh Sidhu (Student ID: 815805), XYZ (Student ID: 1231231), and ABC (Student ID: 3213213) have actively contributed to the development and success of this project. Working together as a group of three, we were able to combine our skills and knowledge to meet the requirements outlined in the assignment.

9. References:

I would like to acknowledge that I used various refences from *lecture slides provided by my instructor, Jeeho Ryoo*, online resources such as *W3Schools* and the official Python documentation to create this assignment.